

AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions of claims in the application:

Listing of Claims:

1. (Currently amended) A system that facilitates the interface of non-integrated applications, comprising:

a processor coupled to memory that retains:

an artifact provider that ~~exposes~~ hosts artifacts of a first ~~of the applications~~ application, the artifacts include items of data the first application publicly exposes to other applications, each artifact is associated with an artifact type, the artifact type can be at least one of a source file, a defect, a requirement, a test result or a build; and

an artifact consumer that exposes at least one reference ~~held by~~ of a second of the applications application, which the reference is a link to at least one of the artifacts of the first application held by hosted by the artifact provider, the link further comprises a link type that describes a relationship between the artifact consumer and the artifact provider.

2. (Original) The system of claim 1, the link is a uniform resource identifier (URI).

3. (Original) The system of claim 1, the artifact provider and the artifact consumer are application program interfaces (APIs) that interface to the respective applications.

4. (Original) The system of claim 1, further comprising a linking component that links the reference with the corresponding artifact.

5. (Original) The system of claim 4, the linking component is an artifact identifier held by the artifact consumer that points to an artifact.

6. (Original) The system of claim 4, the linking component is binary, and is associated with a referring artifact and a referenced artifact.

7. (Original) The system of claim 1, at least one of the provider and the consumer is a tool or service.
8. (Original) The system of claim 1, the artifact provider registers an artifact type for each artifact it provides, and registers a corresponding link type that each artifact can host.
9. (Original) The system of claim 1, further comprising a generic artifact provider (GAP) that interfaces to a tool to facilitate storing and exposing both artifacts and artifact links.
10. (Original) The system of claim 9, further comprising a GAP adapter that provides an interface between the GAP and a non-integrated application.
11. (Original) The system of claim 1, further comprising a cache that stores the artifacts and associated artifact links.
12. (Original) The system of claim 1, further comprising a user interface that facilitates presenting inter-artifact references.
13. (Original) A computer readable medium having stored thereon computer executable instructions for carrying out the system of claim 1.
- 14-16. (Canceled)
17. (Currently amended) The system of claim [[16]] 1, the link is an artifact identifier that is an immutable and uniquely constructed key.
18. (Currently amended) The system of claim [[16]] 1, further comprising a link manager that manages a cache by updating and purging cache contents.
19. (Currently amended) The system of claim [[16]] 1, the artifact provider and artifact consumer are at least one of loosely coupled and tightly coupled.

20. (Currently amended) The system of claim [[16]] 1, further comprising a classifier that makes an inference based on parameters related to at least one of the artifact consumer, artifact provider, and non-integrated applications.

21. (Currently amended) The system of claim [[16]] 1, the artifact provider creates and reveals a URI for at least one of loosely-coupled server-based interactions, loosely-coupled clients, caching, and tightly-coupled interactions that support artifact-specific functions by contract with a caller.

22. (Canceled)

23. (Currently amended) A computer-readable medium having computer-executable instructions for performing a method for facilitating an interface between non-integrated applications, the method comprising:

providing an artifact provider that communicates with a first non-integrated application;

exposing an artifact ~~of the~~ hosted by the first application using the artifact provider, the artifact comprises an item of public data of the first application;

providing an artifact consumer that communicates with a second non-integrated application;

exposing a reference ~~of the~~ held by second application using the artifact consumer; and

linking the reference to the artifact with an artifact identifier.

24. (Original) The method of claim 23, further comprising the acts of:

registering an artifact type for the artifact; and

registering a link type that the artifact hosts.

25. (Original) The method of claim 23, further comprising presenting dependency information of the artifact to a user, the information including at least one of link type, artifact type, artifact name, and modification date.

26. (Original) The method of claim 23, at least one of the artifact consumer and artifact provider is a web service.
27. (Original) The method of claim 23, further comprising generating an artifact proxy that represents data stored in a non-integrated application.
28. (Original) The method of claim 23, the artifact is representative of at least one of a source file, defect, requirement, test result, and build.
29. (Original) The method of claim 23, the act of linking comprising a link that includes a referring URI, a referenced URI, and a link type.
30. (Original) The method of claim 23, further comprising discovering which referring artifacts hold links to a specific referenced artifact.
31. (Original) The method of claim 23, further comprising raising an event when the artifact is at least one of created, deleted, and changed.
32. (Original) The method of claim 23, further comprising providing external addressability for the artifact by the artifact provider.
33. (Original) The method of claim 23, further comprising providing a generic API that is both an artifact provider and an artifact consumer.
34. (Currently amended) A system that facilitates the interface of non-integrated applications, comprising:
a processor coupled to memory, the processor configured to act as:
means for exposing an artifact of a first application;
means for exposing a reference of a second application;
means for linking the reference to the artifact with an artifact identifier; and
means for caching the artifact and the artifact identifier; and with eaching means

means for employing the linked reference to execute predetermined behavior associated with the artifact.

35. (Original) The system of claim 34, further comprising means for synchronizing the caching means with a source of the artifact and a source of the artifact identifier.

36. (Original) The system of claim 34, further comprising means for defining the artifact and artifact identifier in XML.

37. (Original) The system of claim 34, further comprising means for filtering a query.